International Journal of Information Security
https://doi.org/10.1007/s10207-019-00449-8

REGULAR CONTRIBUTION

SonarSnoop: active acoustic side-channel attacks

Peng Cheng[1] · Ibrahim Ethem Bagci[1] · Utz Roedig[2] · Jeff Yan[3]

Abstract
We report the first *active* acoustic side-channel attack. Speakers are used to emit human inaudible acoustic signals, and the echo is recorded via microphones, turning the acoustic system of a smart phone into a sonar system. The echo signal can be used to profile user interaction with the device. For example, a victim's finger movements can be inferred to steal Android unlock patterns. In our empirical study, the number of candidate unlock patterns that an attacker must try to authenticate herself to a Samsung S4 phone can be reduced by up to 70% using this novel acoustic side-channel. The attack is entirely unnoticeable to victims. Our approach can be easily applied to other application scenarios and device types. Overall, our work highlights a new family of security threats.

Keywords Side-channel attack · Acoustic system · Active sonar · Mobile device

1 Introduction

Radar and sonar systems use radio and sound waves to track objects, including humans. In recent years, this technology has been developed extensively to support human–computer interactions by tracking the movement of human bodies, arms, hands or even fingers [19,22]. However, existing work has rarely investigated the security implications of those technologies.

In this paper, we report some alarming security implications of tracking human movement via sound waves. Specifically, we present the first active acoustic side-channel attack[1] that can be used to steal sensitive information such as Android unlock patterns. In our attack, human inaudible acoustic signals are emitted via speakers and the echo is recorded via microphones, turning the acoustic system of a smart phone into a sonar system. The echo stream not only gives us information about a victim's finger movement but leaks her secrets too.

All known acoustic side-channel attacks are passive, meaning that acoustic signals in the side-channel are generated by the victim but are eavesdropped by the attacker. In contrast, our approach is an active side-channel, meaning that acoustic signals in the side-channel are induced by the attacker.

In our experiment, we use an off-the-shelf Android phone as an example of a computer system with a high-quality acoustic system. We repurpose the acoustic system for our side-channel attack. An inaudible signal is emitted via speakers, and the echo is recorded via microphones turning the acoustic system of the phone into a sonar system. Using this approach, an attacker that obtains control over a phone's speaker and microphone is able to observe user interaction, such as the movement of the user's fingers on the touch screen. As the emitted sound is inaudible for the user, it is hard to detect that the sound system is being used to gather information.

To illustrate the capability and potential of this novel active acoustic side-channel attack, we use the task of stealing Android unlock patterns as a case study. We choose this example, since Android is a popular phone OS, and since its unlock patterns are one of the most widely used authentication mechanisms. Our aim is to demonstrate the general viability of our new acoustic side-channel, not to improve the

[1] This paper initially appeared as [8].

✉ Jeff Yan
jeff.yan@liu.se

Peng Cheng
p.cheng2@lancaster.ac.uk

Ibrahim Ethem Bagci
i.bagci@lancaster.ac.uk

Utz Roedig
u.roedig@cs.ucc.ie

[1] Lancaster University, Lancaster, United Kingdom

[2] University College Cork, Cork, Ireland

[3] Linköping University, Linköping, Sweden

Published online: 05 July 2019

specific task of stealing unlock patterns. It would be interesting future research to compare the effectiveness of our approach with older methods and to identify the best method for stealing unlock patterns, but these are beyond the scope of this paper.

It might appear that our contribution is merely another phone-based side-channel, among many of those that have been investigated for smartphones [24]. However, this is a false impression. Although our experiments are carried out with a phone, the method we show is applicable to many other kinds of computing devices and physical environments where microphones and speakers are available. Perhaps more importantly, when examined in the context of acoustic attacks, our work is particularly significant in that it is the first *active* acoustic side-channel to be reported.

Specific contributions of this paper include:

1. *SonarSnoop framework* We establish the first active acoustic side-channel attack and present *SonarSnoop*, the framework of generic value to construct and implement such attacks.
2. *Unlock pattern stealing* We evaluate the performance of SonarSnoop in stealing unlock patterns and show that the number of unlock patterns an attacker must try until a successful authentication can be reduced by up to 70% using the acoustic side-channel. This attack is entirely unnoticeable to a victim; no noise and no vibration are induced.
3. *A family of security threats* We discuss a number of new attack scenarios that extend our experiment setting. We show that SonarSnoop represents a family of new threats.

The next section describes relevant background on phone unlock patterns, the acoustic side-channel and how to exploit it for an effective attack. Section 3 describes SonarSnoop, the system used to spy on user interactions with a phone, discussing in detail the challenging aspects of signal generation and signal processing necessary to reveal user interaction. Section 4 describes our experimental evaluation using a user study. Specifically, we evaluate the effectiveness of different decision-making strategies. Section 5 discusses findings and limitations. Section 6 generalises the SonarSnoop attack and discusses further attack scenarios, potential countermeasures and broader implications of acoustic side-channel attacks. Section 7 describes related work, and Sect. 8 concludes the paper.

2 Stealing phone unlock patterns via acoustic side-channel attacks

Unlock patterns are often used to secure access to Android phones. We investigate a novel active acoustic side-channel attack to steal these patterns. Previous research investigated different methods for stealing unlock patterns, e.g. using smudges [2], accelerometer readings [3] or video footage [30].

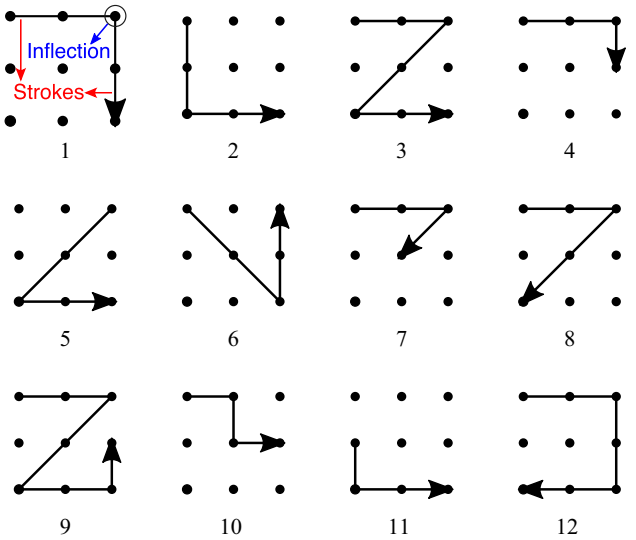

Fig. 1 The twelve most popular unlock patterns according to [10]. An unlock pattern connects a number of dots, and it can be decomposed into several strokes separated by inflections

2.1 Phone unlock patterns

We consider the unlock pattern mechanism available on Android phones. The user is presented with a 3×3 grid of positions. Using the touch screen, the user has to draw a pattern, connecting positions in the correct sequence to authenticate.

Figure 1 shows some examples of such an *unlock pattern*. For the first pattern on the figure, the user has to connect 5 *positions* on the screen in the correct order starting from the top left position. The phone is blocked if the user fails to draw the correct pattern five times in a short period.

The unlock pattern can be decomposed in multiple *strokes* which are separated by *inflections*. Positions that can be reached without changing drawing direction make up one stroke. We use this approach of pattern decomposition later in the paper.

In theory, there are $389,112 \approx 2^{19}$ possible unlock patterns [27]. However, not every pattern is chosen by users with the same probability. Users have bias in choosing unlock patterns, and models have been created to estimate the likelihood of unlock patterns [27]. This bias can be used by adversaries to improve their guess on unlock patterns.

Figure 1 gives the 12 most common unlock patterns in the real world, according to a recent empirical study [10]. In our user study, we focus on stealing these most likely patterns only, for the following reasons. First, unlock patterns have a

highly non-uniform distribution, and those 12 common patterns account for more than 20% of user choices [10]. We aim for these high-priority targets only, just like rational adversaries often choose to do. Second, we aim to make our user study reasonable to participants so that it will not take too much of their time to complete the study. An overly lengthy user study will be tedious and boring, and it will scare away potential participants. In the worst scenario, a bored participant can circumvent the study by producing useless data or otherwise jeopardising our experiment's validity. Third, as mentioned earlier, our purpose is not to steal the most unlock patterns or propose the best experiment of that kind. Instead, our modest aim is to use an experiment to testify the feasibility of our acoustic side-channel attack. We believe our design choice is sufficient for the purpose. Overall, our design choice is not a random decision, but one based on careful deliberation, with multiple factors and their trade-off taken into considerations.

2.2 An acoustic side-channel

The acoustic channel can be used to infer user behaviour using either a passive or active approach. A passive system assumes that the observation target itself emits sound that can be recorded for analysis. An active system uses speakers to emit a sound wave and microphones to collect echo data.

In this work, we use the active approach. The speakers of the system emit an orthogonal frequency division multiplexing (OFDM) sound signal. We use OFDM because it has a good correlation property [19] and it is easy to confine the signal to the higher inaudible frequency range (see Sect. 3.3). The sound signal sits in a frequency range that is inaudible to most people (18–20 kHz). The microphones are used to record the echo. By analysing the recorded echo, it is possible to deduce user interaction patterns with the touch screen.

When using this technique during a user's interaction with the unlock mechanism, information regarding the unlock pattern is leaked which constitutes the acoustic side-channel.

2.3 Threat model

We consider an adversary's goal is to steal a user's unlock pattern. We assume that the adversary uses software deployed on the user's phone to achieve this goal. We further make the assumption that the adversary uses the acoustic system (speakers and microphones) on the phone to achieve this goal. We assume the adversary is able to deploy code on the user's phone which carries out the acoustic side-channel attack.

Typically, such code might be installed in form of an App. The adversary may develop an App that incorporates code to execute the acoustic side-channel attack and presents itself to the user as a benign App such as a Weather App or a Game. The existence of Apps with such hidden malicious functionality in the Android marketplace is well documented [32,33].

The App will require access to microphones. The user will be asked to grant access to this when the application is first launched. Users often grant such access as they rarely question the necessity of these requests [13]. In addition, the App might be designed such that this permission seems reasonable to the user. For example, the App might have sound effects and offer voice control.

To be effective, the App will have to be active when the user enters the unlock pattern. Thus, the App has to be running in the background and become active when the phone starts.

The App may also make use of available communication channels to transport observed data to a back-end system. The back-end system can be used to analyse the acoustic traces, avoiding suspicious heavy computation on the user's phone. Again, such communication falls within normal operational behaviour of Apps and would not raise suspicion.

The acoustic system is specific to the phone model. Different phones provide a different number of speakers and microphones, and they are placed differently. Thus, an active acoustic side-channel attack must be tuned to the model of the phone. We assume that the adversary is able to obtain the same phone model as used by the target in order to adjust signal processing parameters to the target environment.

2.4 Attack success

An adversary is successful if he or she has: (i) deployed malicious code on the target's phone; (ii) collected sufficient data from the acoustic side-channel during the users' interaction with the unlock mechanism; (iii) analysed the data and extracted unlock pattern candidates; and (iv) the number of extracted pattern candidates is smaller than the number of trials the OS allows.

The challenging parts of this attack sequence include collecting useful data from the acoustic side-channel and designing a data analysis framework for inferring unlock patterns. The next sections will focus on these elements. We consider the deployment of malicious code on a target's phone a solved problem, as is the common practice in the literature [32,33].

3 SonarSnoop

This section describes *SonarSnoop*, our framework to execute an acoustic side-channel attack on the Android phone unlock pattern. We call the framework SonarSnoop as we use the acoustic detection to snoop on user interaction, bearing similarities with sonar systems. The system is geared towards unlock patterns; however, by exchanging elements of the sig-

nal processing and decision-making components the system could be repurposed for other side-channel attacks such as observing user interaction with a banking App.

Our work was inspired by FingerIO [19], which was a system for user interaction based on active acoustic sonar. However, FingerIO was used to track movement of gestures in the vicinity of a phone while our system requires to track finger movements on the screen. Thus in SonarSnoop, the close proximity and the fact that the user holds the phone during interactions create additional complexity. We also have to modify both signal generation and processing to tackle our attack scenarios.

The speakers of the phone send an inaudible OFDM sound signal which all objects around the phone reflect. The microphones receive the signal and also the reflections (delayed copies of the signal). The time of arrival of all echoes does not change when objects are static. However, when an object (a finger) is moving, a shift in arrival times is observed. The received signals are represented by a so-called *echo profile matrix* which visualises this shift and allows us to observe movement. Combining observed movement from multiple microphones allows us to estimate strokes and inflections (see Fig. 1). By combining the estimated sequence of observed strokes, we can then estimate the unlock pattern they represent.

The four main components of SonarSnoop are:

- *Signal generation* Using the speakers of the phone an OFDM signal is produced. The signal is inaudible and suitable for close-range tracking of fingers.
- *Data collection* Data are collected via the device's microphones.
- *Signal processing* Echo profiles are created followed by removal of noise and artefacts. Then, features (finger movement direction and distance) are extracted.
- *Decision-making* Using the extracted features, the unlock patterns (represented by their decomposition in strokes and inflections) are discovered. We provide alternative methods to do this.

3.1 Signal generation

Signal generation is based on FingerIO [19] with modifications tailored to our device and application scenario. We introduce some additional processing and filtering steps.

Identical to FingerIO, 48 kHz is used as the sampling frequency. According to Nyquist theorem, this supports a sound wave of up to 24 kHz. This importantly supports frequencies above 18 kHz, which is the highest frequency most adults can hear. A vector comprising 64 subcarriers, each covering 375 Hz, is composed. All subcarriers outside of the intended band (18–20 kHz) are set to 0, and all others are set to 1.

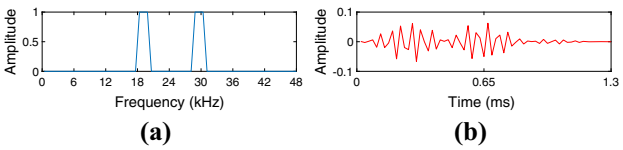

Fig. 2 Sonar signal generation. **a** 128-point vector in frequency domain and **b** 64-point signal in time domain

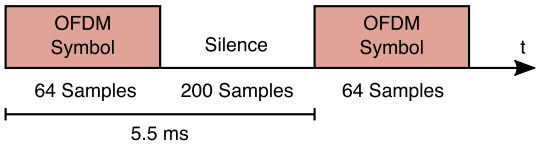

Fig. 3 The sound frame as played continuously over the speakers

The next signal generation steps are in addition to the mechanism used by FingerIO and are used to adjust to our phone and application scenario. A copy of the vector is reverse ordered, and the two vectors concatenated, resulting in the vector shown in Fig. 2a. The 128-sample time domain signal is generated using the inverse fast Fourier transform (IFFT). The real part is divided in half and the first half used as the signal. As this introduces spectral leakage into the audible hearing range, we remove unwanted low frequencies using a Hanning window. The final signal in the time domain is shown in Fig. 2b. The signal is padded with silence to introduce a 264-sample interval and a duration of 5.5 ms. This ensures that all echoes are captured before the next pulse is emitted. The frame is repeated continuously, producing a signal as shown in Fig. 3.

We expect that further optimisation is possible. However, we found the performance to be sufficient for our work.

3.2 Data collection

The phone's microphones are used to record sound data using a sampling frequency of 48 kHz. Noise is introduced by the environment and is recorded together with the received OFDM signal. However, ambient noise does not interfere excessively with the signal processing stage.

3.3 Signal processing

Signal processing comprises (i) echo profile generation, (ii) noise removal and (iii) feature extraction.

Echo profile creation

Echo data are recorded and transformed into an echo profile for each present microphone. The processing for each microphone is the same, except for parameter settings taking into account the positioning of microphones in relation to movement locations. Most modern phones provide at least two

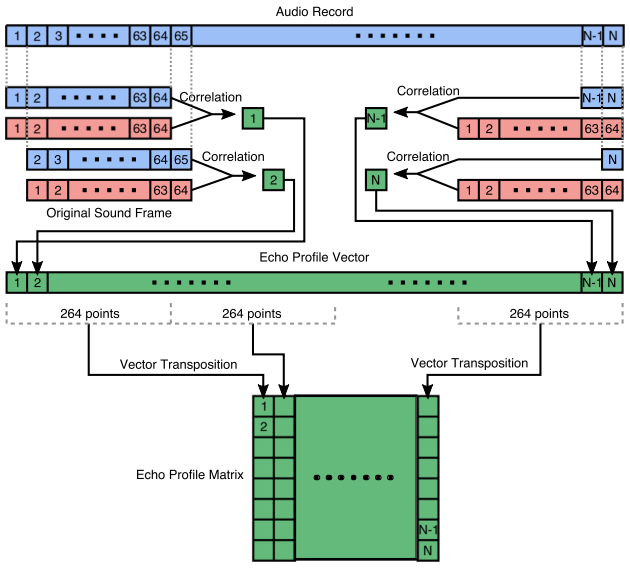

Fig. 4 The process describes the transformation of the audio signal into the echo profile vector and finally the echo profile matrix

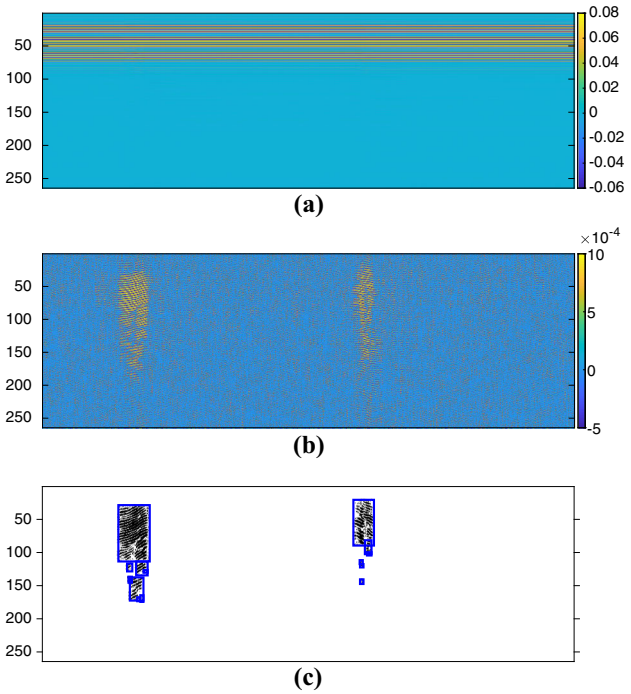

Fig. 5 **a** Raw echo profile matrix. **b** Echo profile matrix after column-wise subtraction. **c** Echo profile matrix after binarisation and segmentation; blue-coloured bounding boxes indicate detected strokes

microphones, one on top of the phone and one on the bottom; we tailor the following process to two microphones, but the described methods can be extended to more microphone inputs.

We create the echo profile by calculating the correlation between the original sound frame and the echo data (see Fig. 4). The original sound frame is a 64-point signal (64 sample points in the time domain over a duration of 1.3 ms). Therefore, we take 64-point sized chunks from the recorded echo data and apply a sliding window, shifting 1 point at a time and calculating the correlation with the original sound frame. Each correlation result is then concatenated to create the *echo profile vector*.

The data emitted on the speakers consist of periodical 264-point long sound frames, and echoes are observed within this period. When there is no object movement, echoes will be observed at the same time within each 264-point frame. When objects move, echo positions will change within each following 264-point frame. This can be visualised by transforming the echo profile vector into an *echo profile matrix*. We take 264-point sized chunks from the echo profile vector and transpose them to create the echo profile matrix. Figure 5a shows an example echo profile matrix. The x-axis and y-axis of the matrix correspond to time and distance, respectively.

When an object moves, slight variations comparing one column of the echo profile matrix to the next can be observed. Depending on the microphone location in relation to the movement and the speed of moving objects, it is necessary to compare column i with column $i + \delta$ to see clear changes. For the phone used in our experiments, we set $\delta = 8$ (44 ms separation) for the bottom microphone and $\delta = 16$ (88 ms separation) for the top microphone. We chose these values as they provided the best performance for our application case. Figure 5b shows an example of the echo profile matrix after subtraction of values in column i and $i + \delta$. The finger movements are now clearly visible and can be analysed by suitable algorithms.

Noise removal

Before analysing data captured in the echo profile matrix, noise is removed. We consider here noise from the sonar system and not ambient sound, because such ambient noise does not correlate with our original signal and therefore does not interfere with the sonar signal analysis. An example result of this clean up procedure is shown in Fig. 5c which corresponds to the data shown in Fig. 5b.

First, we transform the echo profile matrix into a binary matrix by setting values above a threshold to 1 and below to 0. Thus, only correlation above the threshold is taken into account as this corresponds to significant movements. The threshold is chosen as the 94th percentile of all the values in the matrix. We found that this threshold setting performs well in the context of our work.

We use image processing techniques to extract features from the echo profile matrix. Thus, our next step of noise removal is tailored to this method of feature extraction. We use the concept of connected components (CCs) to detect

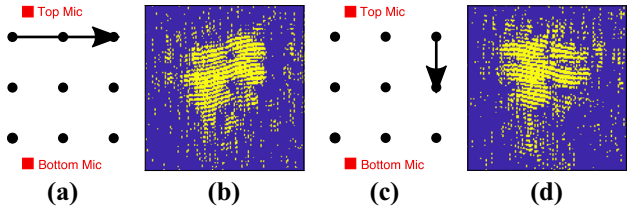

Fig. 6 Two strokes with different direction information. **a** A stroke that is moving away from the bottom microphone. **b** Connected components (CC) of the stroke shown in *a* extracted from the bottom microphone. It shows an ascending trend. **c** A stroke that is moving towards the bottom microphone. **d** CC of the stroke in *c* extracted from the bottom microphone. It shows a descending trend

areas of activity (corresponding to strokes) in the binary echo profile matrix. Each CC is defined as area containing more than 20 connected 1s. We remove all 1s from our echo profile matrix that are not included in such CCs. Again, a threshold of 20 was found to be suitable for our application context.

Feature extraction

We use the CCs to locate areas of movement in the binary echo profile matrix. Each CC is described by a bounding box (BB) which is the smallest rectangle surrounding the CC as shown in Fig. 5c. CCs that identify one stroke are grouped together. For each group of CCs, we extract two features: (i) movement direction and (ii) movement distance. Movement direction relates to the angle of lines visible in the CCs of a group. Movement distance relates to the height of BBs in each group.

Before extracting features, we exclude some CCs. We remove CCs that are only visible on one microphone input; movement should be detected clearly by both microphones at the same time. We also remove overlapping CCs when more than half of the smaller CC overlaps in x and y direction. Thus, the number of CC within a group is reduced, simplifying analysis without loosing accuracy.

CCs are assigned to groups by using a separation of 80 columns (i.e. by 440 ms) on the x-axis. A user pauses between strokes, and we use this separation to group CCs belonging to the same stroke. This method relies on a visible pause at inflections. Other data analysis methods need to be used to determine CC groups when this cue is not present. We did experiments during which a user does not need to pause at each inflection. The results remain similar only except CCs of different strokes connect together. This fact does not invalidate our approach, but additional image processing techniques are in need to separate these CCs.

Objects moving away from a microphone produce an ascending trend in the CC, while objects moving towards produce a descending trend. Figure 6 gives an example of this behaviour pattern; two strokes captured by the bottom

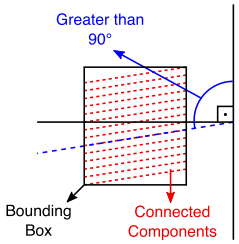

Fig. 7 The relation between the ascending trend of a connected component and the angle (orientation) result of Gabor filter

microphone are shown. The CC in Fig. 6b has an ascending trend, and the CC in Fig. 6d has a descending trend. To identify these trends automatically, we use a Gabor filter [26], which is a well-known linear filter and often used for extracting orientation information. We quantify the orientation (i.e. the angle) of lines within the CC in each BB using Gabor filter. If the angle is greater than 90° as shown in Fig. 7, it means that an object is moving away from a microphone, while an angle smaller than 90° means that an object is moving towards the microphone. After obtaining the angle information of each BB belonging to a stroke, we combine these into a single value. We weigh the angle information of each CC by the size of the BB. In the remainder of the paper, we call this feature representing a stroke's direction the *angle*.

Movement distance can be inferred from the heights of the BBs within a group. As a group corresponds to a stroke in our case, the height of each BB contributes to the movement distance of a stroke. If the stroke is long, the BBs cover more space vertically. In the remainder of the paper, we refer to this feature as the *range*.

3.4 Decision-making

SonarSnoop gathers stroke information via the features angle and range. This information has to be translated into a meaningful user interaction pattern. Depending on the application, very different decision-making processes can be appropriate.

We consider in this paper only the task of stealing phone unlock patterns as described in Sect. 2.3. For this purpose, we define 3 different decision-making options named D1, D2 and D3 which operate very differently.

- **D1** simply uses the features angle and range and classifies each stroke. The resulting sequence of strokes is then the assumed unlock pattern of the user.
- **D2** uses only the angle feature of strokes. The sequence of directions reveals a set of candidate patterns. The set of candidate patterns is likely to contain the user's unlock pattern.
- **D3** combines D2 and D1. First, a set of candidates is determined by investigating the angle feature of strokes.

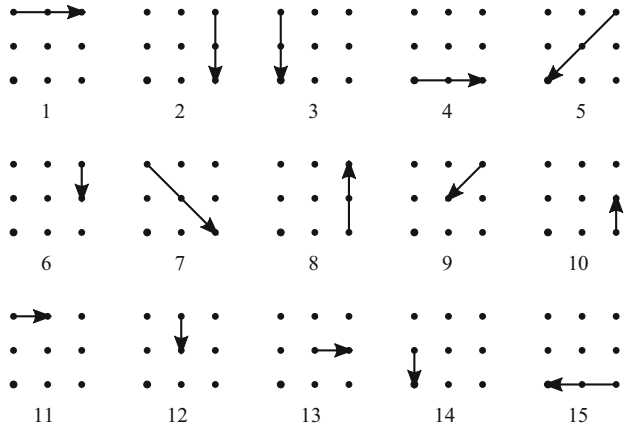

Fig. 8 The 15 strokes used to compose the 12 unlock patterns shown in Fig. 1

Then within the candidate set, angle and range are used to classify strokes and identify the user's pattern.

Users do change unlock patterns infrequently, and once malicious software is deployed on a phone, multiple unlock procedures can be observed. For each observed unlock procedure, the decision-making process provides one or more candidate pattern. All proposed candidate patterns are ordered according to the number of times they were suggested. The position of the user's pattern in the list of suggested patterns determines the effectiveness of the side-channel attack.

D1 - classifying strokes using angle and range

We classify the strokes using machine learning. The sequence of classified strokes reveals the unlock pattern. We use the 12 most likely unlock pattern as shown in Fig. 1 which decompose into 15 unique strokes as shown in Fig. 8. Sample data for each of the 15 strokes from 2 individuals (trainers) are used to train the classifier. Trainers are not subjects of the user study. We use the direction (angle) and distance (range) of the strokes obtained from the echo data of both microphones.

There are 3 variants of this decision-making process: **(D1.1)** using data from both microphones; **(D1.2)** using data from the bottom microphone only; and **(D1.3)** using data from the top microphone only.

D2 - grouping patterns using angle

For this method, we use only the direction (angle) of a stroke, whether it is moving towards a microphone or away. Then, we look at the combination of the strokes to guess the pattern. For example, the first stroke of the Pattern 1 in Fig. 1 is *moving away* from the bottom microphone and the second stroke is *moving towards* the bottom microphone.

Table 1 Groups of patterns that have the same number of strokes with the same behaviours when using one microphone. Behaviours are shown as *A* if the stroke is moving away from the microphone, and *T* if the stroke is moving towards to the microphone

Patterns	Bottom mic.	Patterns	Top mic.
1, 4, 7, 8	A - T	1, 2, 4, 5, 8, 11	A - A
2, 5, 6, 11	T - A	3, 10	A - A - A
3, 10	A - T - A	6, 7	A - T
9	A - T - A - A	9	A - A - A - T
12	A - T - T	12	A - A - T

Table 2 Groups of patterns that have the same number of strokes with the same behaviours when using both microphones. Behaviours are shown as *A* if the stroke is moving away from the microphone, and *T* if the stroke is moving towards to the microphone

Patterns	Bottom microphone	Top microphone
1, 4, 8	A - T	A - A
2, 5, 11	T - A	A - A
3, 10	A - T - A	A - A - A
6	T - A	A - T
7	A - T	A - T
9	A - T - A - A	A - A - A - T
12	A - T - T	A - A - T

Pattern 4 , Pattern 7 and Pattern 8 have the same behaviour from the perspective of the bottom microphone. Therefore, using only angle information, a group of patterns is identified. Table 1 shows pattern groups for each microphone that have the same stroke behaviour when considering patterns as shown in Fig. 1.

The group sizes can be reduced by considering data from both microphones together. For example, the first stroke of the Pattern 1 in Fig. 1 is *moving away* from the bottom microphone and the second stroke is *moving towards* the bottom microphone; considering the top microphone the first and second stroke are *moving away* from microphone. Only Pattern 4 and Pattern 8 have this same behaviour. Table 2 shows pattern groups that have the same stroke behaviour.

It may happen that the analysis of stroke patterns using angle information of both microphones is inconclusive. For example, the strokes from the top microphone are reported as *moving towards* and *moving towards*, and the strokes from the bottom microphone are reported as *moving away* and *moving towards*. In this case, no group mapping exists as the combination cannot be mapped to any entry in Table 2. In such situation, where no match is possible, we choose to fall back on data collected from one microphone.

We use four strategies to operate this decision-making process: **(D2.1)** using data from both microphones and using only the bottom microphone in inconclusive situations;

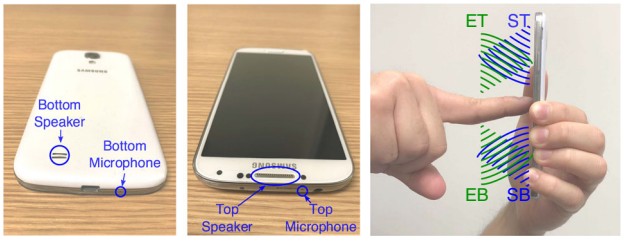

Fig. 9 Galaxy S4 used in the experiments. **a** Location of the bottom speaker and the bottom microphone. **b** Location of the top speaker and the top microphone. **c** Simplified reflection paths of the bottom speaker (SB), the top speaker (ST), the echo coming to the bottom microphone (EB) and the echo coming to the top microphone (ET)

(D2.2) using data from both microphones, using only the top microphone in inconclusive situations; **(D2.3)** using data from the bottom microphone only; **(D2.4)** using data from the top microphone only.

D3 - grouping patterns using angle and classifying strokes using angle and range

We combine the first two approaches to improve the overall accuracy. We first use method D2 to identify a pattern group, and then, we select a specific pattern from this group using method D1. This approach improves on using D1 alone as the pool of candidate patterns is reduced before machine learning is applied. We train machine learning models for the strokes of each group using corresponding microphone's data.

Similar to method D2, four different operation modes can be used: **(D3.1)** using data from both microphones and using only the bottom microphone in inconclusive situations; **(D3.2)** using data from both microphones, using only the top microphone in inconclusive situations; **(D3.3)** using data from the bottom microphone only; **(D3.4)** using data from the top microphone only.

4 Experimental evaluation of SonarSnoop

We evaluate SonarSnoop using a Samsung Galaxy S4 running Android 5.0.1. A dedicated evaluation App is used for a user study to prompt users to input unlock patterns which we then aim to reveal using SonarSnoop.

4.1 User study

The Samsung Galaxy S4 provides two speakers and two microphones as shown in Fig. 9. We execute the data collection component of SonarSnoop on the phone. Signal generation, signal processing and decision-making are executed on a dedicated PC.

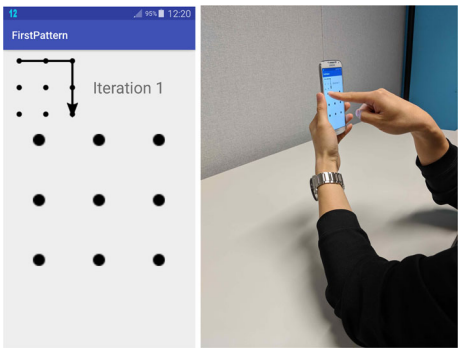

Fig. 10 **a** Screenshot of the App used for our user study. **b** An example demonstration of the user study

For the experiments, we develop a dedicated evaluation App instead of using the built-in pattern unlock mechanism of Android. The App replicates the pattern unlock mechanism and provides additional features to simplify evaluation. The App provides the user with a 9-point matrix to enter unlock patterns. In addition, the user interface shows the pattern we expect the user to draw. The user interface is shown in Fig. 10a. The evaluation App guides a user through the experiment and ensures that echo data recorded by SonarSnoop can be matched with the pattern the user was asked to draw (ground truth).

We ran a user study with 10 volunteers (we obtained approval for the study from the University Ethics Committee). Each volunteer was asked to draw each of the 12 unlock patterns as shown in Fig. 1 five times. The evaluation App guided the volunteers through this process which took up to 30 min.

The participant is asked to hold the phone with one hand and to draw the pattern with the other. The participants sat at a table and rest the arm holding the phone with their elbow on the table (see Fig. 10b).

All experimentation was carried in an open plan office without restrictions on the environment noise. During our experiment, usual office noise was present (people moving, chatting, moving chairs, opening doors). The results shows that our approach is fairly robust against such environment noise.

4.2 Evaluation metrics

Using the data collected in our user study, we evaluate the three different variants of our decision-making process. To judge performance, we use five key metrics:

– Pattern guess rate per user **(M1)**: For each user, we calculate the ratio of successfully retrieved patterns to the number of patterns in the pool (i.e. 12). A pattern is suc-

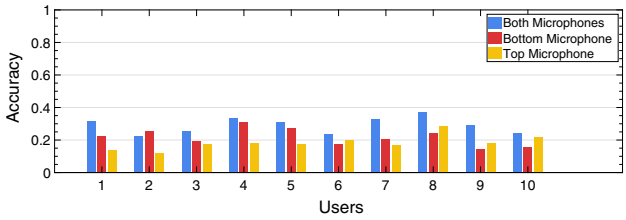

Fig. 11 Classification accuracies for 15 strokes shown in Fig. 8 for 10 users

cessfully retrieved if the decision-making suggests a set of patterns which contains the correct one.

- Pattern candidates per user **(M2)**: Without the aid of our side-channel attack, the attacker must perform a random guess (i.e. selecting randomly from a pool of 12 patterns in our case). This metric describes the average size of the pattern pool after the decision-making per user. If the size of the remaining pattern candidate pool is reduced, it will improve pattern guess rate.
- Pattern guess rate per pattern **(M3)**: This metric is similar to M1. However, the rate of success is per pattern instead of per user. The metric shows how successful a specific pattern is retrieved across all users within our study.
- Pattern candidates per pattern **(M4)**: This metric is similar to M2. Here the average size of the pattern pool after decision-making is calculated per pattern across all users in the study.
- Attack attempts **(M5)**: This metric is based on M2. However, the unlock patterns are ordered by the number of times they were suggested. This gives the sequence of patterns an attacker will try. M5 is the position of the user's pattern in this ordered pattern pool.

4.3 Decision-making option D1

With this decision-making variant, we classify the individual strokes of the patterns and then deduce the pattern from the result (see Sect. 3.4). Figure 8 shows the 15 unique strokes of the 12 patterns shown in Fig. 1 which are elements of our study. We train our machine learning model using data obtained from 2 trainers. We collect between 30 and 40 samples in total for each stroke. We test various algorithms using fivefold cross-validation on the training data and pick *Medium Gaussian SVM* algorithm with *kernel scale* parameter 2.2, as it performs best among other algorithms in terms of accuracy.

Figure 11 shows the classification accuracies for the 15 strokes shown in Fig. 8 for 10 users that participated in our study. The figure shows accuracies for each user with different combinations of feature sets. Using both microphones gives the best performance as we would expect. The highest accuracy value of 0.37 is achieved with User 8 when using

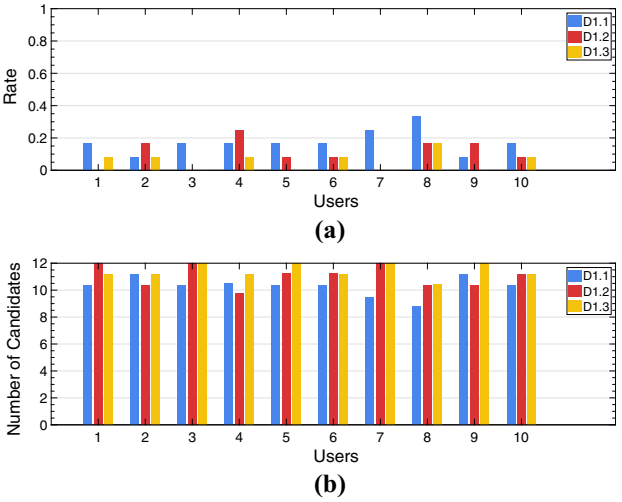

Fig. 12 Results per user with decision method D1. **a** Pattern guess rate (Metric M1). **b** Average number of pattern candidates remained after predictions (Metric M2)

both microphones. We obtain the best overall average accuracy when using both microphones (accuracy of 0.29).

Next we combine the classified strokes to guess the users' pattern. Figure 12a shows the pattern guess rate per user (Metric M1). The rate is shown using variation D1.1, D1.2 and D1.3 of our decision-making method D1. As a reflection of the results shown Fig. 11, we obtain the best rate of 0.33 for User 8 when using data from both microphones, and the best average value of 0.18 across all users is also achieved when using both microphones.

Figure 12b shows the average number of candidate patterns remained after predictions for each user (Metric M2). A minimum number of candidates of 8.83 is achieved when using both microphones for User 8, and we obtain the minimum average value of 10.28 when using both microphone across the user population.

Figure 13a shows pattern guess rate across all users for each pattern (Metric M3). Although the average rate of 0.18 is achieved when using both microphones, Pattern 5 is revealed for 9 users within 5 iterations.

Figure 13b shows the average number of candidates remained after predictions for each pattern (Metric M4). The minimum value of 3.20 is achieved with Pattern 5 when using both microphones. We obtain the minimum average value of 10.28 when using both microphones.

Summary The results show that method D1 reduces the candidate pool of patterns (Metrics M2, M4). Thus, we show that the acoustic side-channel is generally useful to an attacker. However, the improvement is not very significant. The average number of candidate patterns for the attacker to try is reduced from 12 to 10.28 (Metrics M2, M4).

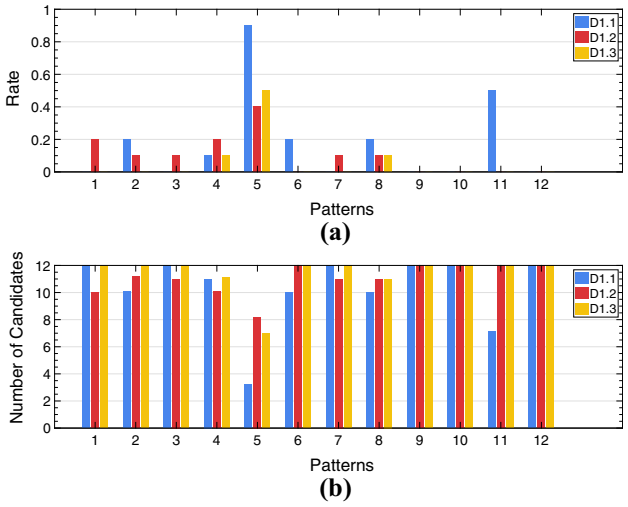

Fig. 13 Results per pattern with decision method D1. **a** Pattern guess rate (Metric M3). **b** Average number of pattern candidates remained after predictions (Metric M4)

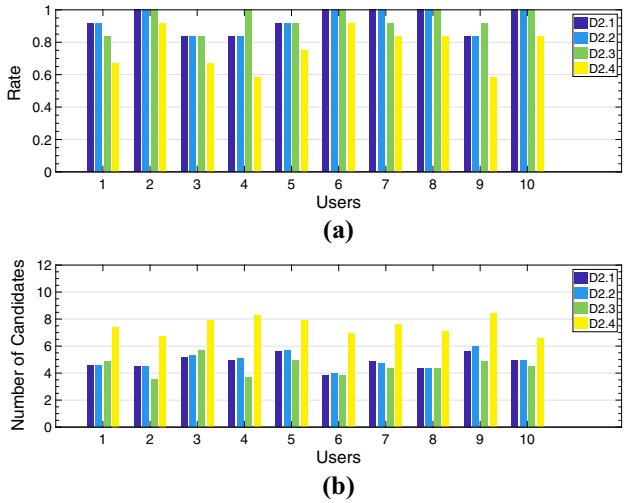

Fig. 14 Results per user with decision method D2. **a** Pattern guess rate (Metric M1). **b** Average number of pattern candidates remained after predictions (Metric M2)

4.4 Decision-making option D2

With this decision-making variant, we identify candidate groups based on the angle information. Some patterns share the same number of movements with the same behaviours (moving away or moving towards), and we cannot narrow the decision down to a single pattern.

Table 2 shows groups of patterns that have the same number of movements with same behaviours when using both microphones.

The rates (Metric M1) as shown in Fig. 14a are above 0.83 for all users when using both microphones (D2.1 and D2.2). The patterns of Users 2, 6, 7, 8 and 10 are mapped to their

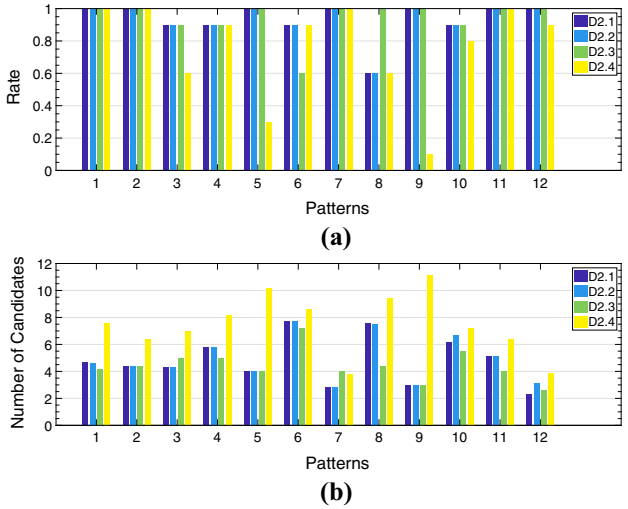

Fig. 15 Results per pattern with decision method D2. **a** Pattern guess rate (Metric M3). **b** Average number of pattern candidates remained after predictions (Metric M4)

groups shown in Table 2 with 100% success when using both microphones, and the best average value of 0.93 for M1 is achieved when using both microphones (D2.1 and D2.2).

The minimum number of candidates as shown in Fig. 14b (Metric M2) of 3.50 is achieved for User 2, and we obtain the minimum average value of 4.44 when using only bottom microphone's data (D2.3).

Most of the patterns are mapped to their correct groups (Metric M3 shown in Fig. 15a), and an overall average value of 0.93 is achieved when using the data from both microphones (D2.1 and D2.2).

The average number of candidates remained after predictions for each pattern is shown in Fig. 15b (Metric M4). A minimum value of 2.30 is achieved for Pattern 12 when using both microphones and the bottom microphone is dominant (D2.1). We obtain the minimum average value when using only the bottom microphone (D2.3) which is 4.44.

Summary D2 performs significantly better than D1. The number of pattern candidates is significantly reduced from 12 to 4.4 (Metrics M2, M4). This is an interesting result as this method uses only one feature (angle) for the decision-making. However, the attacker still has more than one candidate due to similarities of patterns and their grouping.

4.5 Decision-making option D3

Here we first map a pattern into a group of patterns by looking at the directions using method D2. The results are then narrowed down further by classifying the unique strokes of the patterns within a group. For this classification, machine learning models for each group are required. For method D1, we had 15 different strokes that need training for the decision-

Table 3 Groups of unique strokes to be trained for each pattern group when using both microphones, only bottom microphone and only top microphone

Both microphones		Bottom microphone		Top microphone	
Patterns	Strokes	Patterns	Strokes	Patterns	Strokes
1, 4, 8	2, 5, 6	1, 4, 6, 8	2, 5, 6, 9	1, 2, 4, 5, 8, 11	1, 3, 5, 14
2, 5, 11	3, 5, 14	2, 5, 6, 11	3, 5, 7, 14	1, 2, 4, 5, 8, 11	2, 4, 5, 6
3, 10	1, 11	2, 5, 6, 11	4, 8	3, 10	1, 11
3, 10	5, 12	3, 10	1, 11	3, 10	5, 12
3, 10	4, 13	3, 10	5, 12	3, 10	4, 13
		3, 10	4, 13	6, 7	1, 7
				6, 7	8, 9

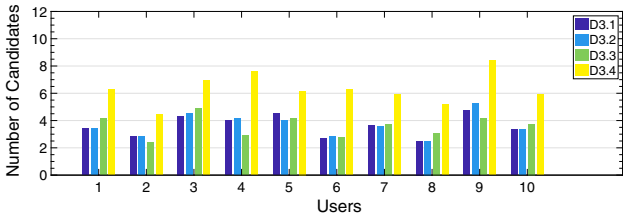

Fig. 16 Average number of pattern candidates remained after predictions for each user (Metric M2) with decision method D3

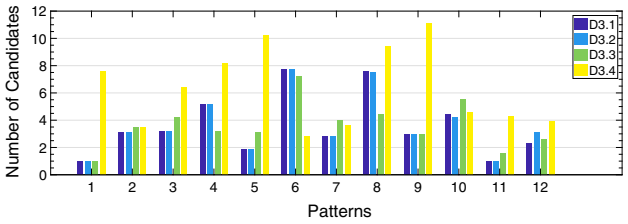

Fig. 17 Average number of pattern candidates remained after predictions for each pattern (Metric M4) with decision method D3

making method D1. For the method D3, the number of unique strokes within each group is less than 15, which will be less challenging for the machine learning models. Therefore, we expected that the algorithm performance will be better than the one in Sect. 4.3.

The groups of patterns that have the same number of strokes with the same behaviours are shown in Table 2. These groups are created using the data of both microphones. For the method D3, we sometimes use only one of the microphone's data as fallback. Therefore, we need to take into account the patterns that are grouped together when using only one of the microphone's data, which are shown in Table 1. We analyse the patterns within each group shown in Tables 1 and 2, then filter out the common strokes of all patterns within this group and only list the unique strokes within each group. For instance, Strokes 2, 5 and 6 are unique for each pattern (Patterns 1, 4 and 8) of the first group shown in Table 2. Unique stroke groups to be trained for each pattern group when using both microphones, only bottom microphone and only top microphone are shown in Table 3. We create machine learning models for the strokes of each group using the corresponding microphone's data. Various machine learning algorithms are applied to training data using fivefold cross-validation, and *Medium Gaussian SVM* algorithm with *kernel scale* parameter 2.2 is chosen for decision-making over users' data.

D3 is an extension of D2, and therefore, the rates achieved (Metric M1 and M3) are the same for D2 and D3. However, the additional processing after identifying candidate sets reduces the pattern candidate sets further (Metric M2 and M4). We therefore present next the results regarding M2 and M4.

The average number of candidates remained after predictions for each user (Metric M2) is shown in Fig. 16. A minimum number of candidates of 2.5 is achieved for User 8. We obtain the minimum average value of 3.6 for D3.1 when looking across all users.

Figure 17 shows the average number of candidates remained after predictions for each pattern (Metric M4). The minimum number of candidates of 1 is achieved for Patterns 1 and 11 when using the data from both microphones (D3.1 and D3.2), which means we just need one attempt to guess these two patterns. We obtain the minimum average value of 3.6 using method D3.1.

Summary Using this method, we can reduce the pattern candidate pool in some cases from 12 to 1 (Metric M4). When looking across all patterns, this method D3 improves on D2. The number of pattern candidates is reduced from 4.4 to 3.6 (Metric M2 and M4). Moreover, the average attack attempt value of 2.71 is achieved when using method D3.2 (Metric M5).

5 Discussions

Our experimental evaluation shows that the active acoustic side-channel is in principle a useful instrument for revealing user interaction patterns on phones. However, our study

and the components of SonarSnoop have limitations and improvements are possible.

5.1 Algorithm performance

D1 is the most generic method, identifying individual strokes and composing these into patterns. The method helps to reduce an attackers effort in guessing the unlock pattern. However, the method does not yield very good results. The average number of candidate patterns that the attacker has to try is reduced from 12 to 10.28.

D2 is much better and provides an average reduction from 12 to 4.4 patterns. This result is achieved by analysing less features from the collected sound data than method D1 (based only on direction of finger movement). However, D2 requires us to decide on a pool of patterns beforehand. This is a limitation D1 is not bound to; however, in practice this may not be problematical as the pool of likely patterns is known [10].

D3 improves on D2 by combining D1 and D2. Patterns are grouped, and thereafter, the method used in D1 is applied to narrow down the pattern candidate pool further. D3 provides an average reduction from 12 to 3.6 pattern (D2 reduces these from 12 to 4.4). Although D3 requires reasonably more computational effort, it gives better results than D2.

5.2 Limitations and improvements

The acoustic signal generation can be improved. We believe it is possible to reduce the silence period in between pulses to achieve better resolution. The current gap size between pulses ensures reflections can be received from up to 1m distance before the next pulse is emitted. Given that we are interested in movement on the screen in close proximity, we can reduce this gap. Also, different signal shapes might be possible that improve system performance.

For convenience and simplicity, we do not implement the system to cope with different users interaction speeds. We use a fixed column width of the echo profile matrix to determine whether there is movement. We calibrated the system to work well with most users. However, if a user draws a pattern very slowly, the differential echo profile matrix may not reveal movements, since the rate of change is too slow to be detected. An improved implementation could support an adaptive feature to adjust with vastly different interaction speeds. Nevertheless, our method is applicable for practical scenarios since we have observed that people draw patterns consistently fast.

We proposed three decision-making strategies. The algorithms sufficiently demonstrate that the active acoustic side-channel attack is feasible. However, we believe it is possible to design better decision-making strategies. For example, additional features could be extracted from the recorded sound data to provide a better basis for decisions. Also, different methods for analysing the existing features (angle and direction) are possible.

SonarSnoop in its current form relies on clear separation of strokes within a pattern. When users do not pause at an inflection, it is currently impossible to distinguish individual strokes. In our user study, we asked users to pause at inflections. We aim to extend the system with methods for automatic separation of strokes. This can be achieved by analysing angle changes within individual connected component (CC).

Our user study has limitations. We had 10 users that were asked to draw 12 patterns 5 times. While the study provided sufficient data to analyse SonarSnoop, it would be useful to expand the data set, e.g. with a greater variety of patterns and with these entered more than 5 times by the users.

6 Attack generalisation and countermeasures

SonarSnoop can do more than stealing unlock patterns on phones. This approach can be applied to other scenarios and device types, and SonarSnoop represents a new family of security threats.

6.1 Attack generalisation

SonarSnoop can be expanded to support different interactions and device types.

SonarSnoop can be extended to observe different user interactions such as gestures or typing on a phone's virtual keyboard. Recognising simple gestures (such as swipe left or right as used for Tinder) would be relatively simple to discern while identifying different key presses on a keyboard is more challenging. Adaptation to different interaction types will enable new side-channel attacks on specific applications.

Our experiment observes user interactions with a touch screen. However, SonarSnoop can be extended to observe user behaviour in some distance to the phone. For example, FingerIO has used a similar approach to observe gestures a meter away from the speaker/microphone. Thus, a phone could be used to observe user interaction with a device (e.g. an ATM) other than the phone itself.

In our study, acoustic emitter and receiver are located in the same device, and situations where these two components are separate should be considered. It is not uncommon that phones are just put aside people's laptops when they work. In this case, speakers on the phone can act as emitter while microphones on the laptop can work as receiver, or vice versa. Similarly, devices do not need to be limited only to phones and laptops. Any devices with microphones and speakers such as tablets and phones, smart watches, cameras or voice assistants are candidates.

6.2 New attack scenarios

We envisage a number of new attack scenarios that extend our experiment.

Stealing personal preferences: Tinder, the popular social search App, helps strangers to socialise with each other. It supports a filter mechanism that two people can only start chatting if they both like each other's profile picture. Tinder treats a user's 'right swipe' actions as like and 'left swipe' as dislike. These swipe actions can be easily differentiated by SonarSnoop.

More and more human gestures are incorporated into the so-called natural user interaction with various computing devices. Our Tinder attack suggests numerous new possibilities for stealing people's sensitive personal preferences via spying on their gestures.

Combo attacks: SonarSnoop can be extended to use additional sensor inputs to boost performance. The combination of multiple sensing inputs has been used successfully in the past. For instance, Simon et al. make use of the front phone camera and microphone recording to infer personal identification numbers (PINs) [23]. They use the front camera to record a video when people input, by tapping on the screen, PINs. The recorded acoustic signal helps to identify frames in which a PIN is entered. Machine learning is used to identify the pressed number in the identified frames. Narain et al. combine a gyroscope and microphone to unveil PINs [21]. Sound and gyroscopes data are used to detect finger tap location on the virtual keyboard or PIN pad.

SonarSnoop can be augmented similarly. For example, data from sensors such as gyroscopes, accelerometers or cameras could be combined with the active sonar approach. It is also possible to use a combination of approaches based on the acoustic channel. Specifically, active and passive approaches can be combined. If passive and our active acoustic side-channel analyses are combined, tapping information (timing and location) and finger movement information (movement distance and movement direction between taps) can be extracted. Such more fine-grained data collection will allow us to infer user interaction with greater detail.

Espionage: Installing hidden acoustic bugs say in an embassy has been a common practice in the intelligence community. This old-fashioned eavesdropping method, when combined with SonarSnoop, will have new advantages. First, the combined use turns a passive eavesdropping into an active one. Second, cautious people know the necessity of playing loud music or otherwise introducing background noise to mitigate the eavesdropping bugs. However, this common countermeasure does little to defend against SonarSnoop, since it is robust to ambient noise.

6.3 Countermeasures

The main feature that enables SonarSnoop is the transmission and reception of inaudible sound. Different hardware and software solutions are possible to interfere with this feature and to prevent an acoustic active side-channel attack.

Sound system design: Devices could be constructed such that transmission of inaudible signals is simply impossible. After all, the intended purpose of a speaker system is to communicate with people who should be able to hear the sound. Supporting the very high frequency range might be useful for high-quality sound systems (e.g. concert halls) but is perhaps unnecessary for simple appliances (e.g. phones or smart TVs). The frequency range that the hardware supports can be restricted to mitigate the threat of SonarSnoop, but this is not viable for already existing systems.

Sound notification: Software or hardware can be used to notify users of a present sound signal in the high frequency range. Users can be alerted by an LED or by a pop-up notification. This can enable users to realise an active side-channel's presence.

Jamming: Another option is to actively disable side-channels. Jamming can actively render side-channels useless to an attacker. For example, Nandakumar et al. proposed to jam acoustic signals in the inaudible range [20]. A device can be designed to monitor acoustic channel activities and, once a threat situation is detected, enable jamming. Alternatively, application software can actively generate noise within the acoustic channel when sensitive tasks are executed (e.g. when a banking App requests a PIN).

Sound system off switch: A sound system (or either microphones or speakers individually) could be disabled during sensitive operations. Either the device provides features that allows software to disable the sound system when needed or a method is provided that allows the user to disable it. For example, a device could provide a switch (the equivalent to a mechanical camera cover) to enable users to control the capability of the device.

Among these countermeasures, no single method fits in all situations. Probably, a stand-alone appliance which can jam in the inaudible frequency range has a best defence capability. However, this approach might not be very user friendly as people need carry an extra device with them.

6.4 Wider impact

A core attacker activity is to study user interaction with systems. The simplest approach here is to follow a victim and observe their actions, for example, to observe a victim entering a PIN code at an ATM. However, people are quite aware

that this might happen and take precautions. An attacker therefore may use a more covert approach and may use a camera for observation. Either a camera is deployed for this purpose or an existing camera is repurposed for this task. For example, the attacker places a camera on the ATM or uses existing CCTV equipment. However, people have also become aware of this attacker approach and are cautious. It is common practice to cover camera lenses on a laptop with a sticker and to be aware of cameras when performing sensitive tasks.

Most devices, including numerous IoT systems, have nowadays a high-end acoustic system. Phones, smart TVs, voice assistants such as Amazon Echo and Google Home have multiple high-end speakers and microphones incorporated. As our study has demonstrated, it is possible to use these systems to gather very detailed information on user behaviour. The information is not yet as detailed as what is possible with optical systems but sufficient to obtain very detailed behaviour profiles. Users are not aware of the capability of sound systems. You would not consider that the presence of a sound system is problematic when carrying out sensitive tasks. People may be wary that conversations are recorded, but they certainly lack awareness that the sound system can be used for observation of movements.

Clearly, this type of threat should be considered. People need to be made aware and the threat should be considered when designing systems.

7 Related work

Our work is the first to propose an active acoustic side-channel attack. It can be a side-channel on phones and on other computing devices where speakers and microphones are available. Closely related work can be divided into three categories. The first category investigates side-channels on phones and wearable devices. The second category explores acoustic side-channel attacks. The third category aims to achieve device-free tracking via acoustics, using the existing speaker and microphones in mobile devices.

7.1 Side channels on mobile devices

A large body of work on side-channel attacks exists, exposing user data via readings of sensors on phones and wearable devices. For instance, work exists on revealing user data, e.g. PIN or graphical passwords through reading of accelerometer data [3,17,18]. Spreitzer et al. [24] provide a comprehensive survey and systematic analysis of this line of work. It clearly supports our claim that our work is the first active acoustic side-channel attack.

7.2 Acoustic side-channel attacks

Obtaining information via an acoustic side-channel is not new. However, existing work mostly utilises acoustic signals passively.

Backes et al. [4] recover the printed content of a dot-matrix printer by analysing printing noise. Faruque et al. [12] reconstruct the object printed by a 3D printer via the emitted sound. Hojjati et al. [15] demonstrate attacks on a 3D printers and a CNC mills using audio and magnetometer data. Other similar work in the manufacturing space is detailed in [5,9]. Toreini et al. [25] decode the keys pressed on an Enigma machine by analysing the emitted sound. Genkin et al. [14] reveal 4096-bit RSA keys using acoustic signals generated by a computer.

There is another group of study focusing on recovering keystrokes on physical or virtual keyboards via acoustics. Cai et al. [7] surveyed potential attacks using microphones. Asonov et al. [1] present an acoustic attack to classify key presses on a physical keyboard. Zhuang et al. [34] further improved this work. Berger et al. [6] present a dictionary attack based on keyboard acoustic emanations. Compagno et al. [11] infer text typed on a keyboard through the acoustic signal captured via Skype chat. Liu et al. [16] use two microphones on a smartphone to inference presses on a keyboard. Narain et al. [21] use a gyroscope and microphones on a mobile phone to predict keystrokes on the virtual keyboard.

All these were acoustic side-channel attacks, but they did not use an active sonar system as we do.

7.3 Device-free acoustic tracking

There is a wealth of work achieving human tracking through RF signals. For example, Wilson et al. [29] use Wi-Fi for tracking. There is existing work in the HCI area focusing on using acoustic signals for tracking finger/hand movements without extra devices [19,28,31].

Nandakumar et al. [19] are related to our work; we base our sonar signal generation on their work. Nandakuma et al. [20] examined the security (more accurately, privacy) implication of tracking human movements with acoustics in a recent study. However, its security and privacy implication was largely a single-bit information, e.g. whether someone was in a room or not, or whether she was moving or standing still. This creates an effective covert channel leaking people's privacy information, but it barely constitutes a side-channel attack.

Our work is different from the existing work in two main aspects. First, the application scenario is different; existing work does not study finger movement on the screen of the phone. Second, existing work does not explore the possibility of stealing sensitive data from users via the acoustic system.

8 Conclusion

We have developed a novel acoustic side-channel attack. Unlike the prior approaches where acoustic signals are passively created by a victim user or computer, we repurpose a computer device's acoustic system into a sonar system. Thereby, an attacker actively beams human inaudible acoustic signals into an environment. The echo stream received not only allows the attacker to stealthily observe a user's behaviour, but also creates a side-channel that leaks her security secrets.

With this active acoustic side-channel, our attack could significantly reduce the number of trials required to successfully guess a victim's unlock pattern on an Android phone. We have noted that attackers do not have to limit themselves to use only smartphones. Instead, our attack appears to be applicable in any environment where microphones and speakers can interact in a way that is similar to our experimental setting.

Thus, our work starts a new line of inquiry, with fertile grounds for future research. For example, it is interesting to investigate and qualify the effectiveness of our attack in different scenarios, and to explore the best countermeasures for each of the scenarios. We also expect our work to inspire novel attacks in the future.

While it helps to improve user experience by tracking human movements and gestures via sound waves or the like, this approach can have a significant security consequence. Unfortunately, this lesson had been largely ignored in the previous research for long. Because of the growing popularity of these invisible 'sensing' technologies, the lesson we have learned here is significant.

Acknowledgements We thank Laurent Simon, Ilia Shumailov and Ross Anderson (all affiliated with Cambridge University) and Chi Zhang (Systems Engineering Research Institute, CSSC) for valuable discussions and suggestions. This work was conceived by JY, and jointly supervised by UR and JY.

Funding JY was supported in part by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.

Compliance with ethical standards

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. This article does not contain any studies with animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Asonov, D., Agrawal, R.: Keyboard acoustic emanations. In: Proceedings of the IEEE Symposium on S&P'04 (2004)
2. Aviv, A.J., Gibson, K., Mossop, E., Blaze, M., Smith, J.M.: Smudge attacks on smartphone touch screens. In: Proceedings of the Usenix WOOT'10 (2010)
3. Aviv, A.J., Sapp, B., Blaze, M., Smith, J.M.: Practicality of accelerometer side channels on smartphones. In: Proceedings of the ACSAC'12 (2012)
4. Backes, M., Dürmuth, M., Gerling, S., Pinkal, M., Sporleder, C.: Acoustic side-channel attacks on printers. In: Proceedings of the USENIX Security'10 (2010)
5. Belikovetsky, S., Solewicz, Y., Yampolskiy, M., Toh, J., Elovici, Y.: Detecting cyber-physical attacks in additive manufacturing using digital audio signing. arXiv preprint arXiv:1705.06454 (2017)
6. Berger, Y., Wool, A., Yeredor, A.: Dictionary attacks using keyboard acoustic emanations. In: Proceedings of the CCS'06 (2006)
7. Cai, L., Machiraju, S., Chen, H.: Defending against sensor-sniffing attacks on mobile phones. In: Proceedings of thr MobiHeld'09 (2009)
8. Cheng, P., Bagci, I. E., Roedig, U., Yan, J.: SonarSnoop: Active Acoustic Side-Channel Attacks. arXiv preprint arXiv:1808.10250. (2018)
9. Chhetri, S.R., Canedo, A., Faruque, M.A.A.: Confidentiality breach through acoustic side-channel in cyber-physical additive manufacturing systems. ACM Trans. Cyber Phys. Syst. **2**(1), 3 (2018)
10. Cho, G., Huh, J.H., Cho, J., Oh, S., Song, Y., Kim, H.: SysPal: System-guided pattern locks for android. In: Proceedings of the IEEE Symposium on S&P'17 (2017)
11. Compagno, A., Conti, M., Lain, D., Tsudik, G.: Don't Skype & Type!: Acoustic eavesdropping in voice-over-IP. In: Proceedings of the ASIA CCS'17 (2017)
12. Faruque, A., Abdullah, M., Chhetri, S.R., Canedo, A., Wan, J.: Acoustic side-channel attacks on additive manufacturing systems. In: Proceedings of the ICCPS'16 (2016)
13. Felt, A.P., Ha, E., Egelman, S., Haney, A., Chin, E., Wagner, D.: Android permissions: user attention, comprehension, and behavior. In: Proceedings of the SOUPS'12 (2012)
14. Genkin, D., Shamir, A., Tromer, E.: RSA key extraction via low-bandwidth acoustic cryptanalysis. In: International Cryptology Conference, Springer, pp 444–461 (2014)
15. Hojjati, A., Adhikari, A., Struckmann, K., Chou, E., Tho Nguyen, T.N., Madan, K., Winslett, M.S., Gunter, C.A., King, W.P.: Leave your phone at the door: side channels that reveal factory floor secrets. In: Proceedings of the CCS'16 (2016)
16. Liu, J., Wang, Y., Kar, G., Chen, Y., Yang, J., Gruteser, M.: Snooping keystrokes with mm-level audio ranging on a single phone. In: Proceedings of the MobiCom'15 (2015)
17. Maiti, A., Armbruster, O., Jadliwala, M., He, J.: Smartwatch-based keystroke inference attacks and context-aware protection mechanisms. In: Proceedings of the ASIA CCS'16 (2016)
18. Marquardt, P., Verma, A., Carter, H., Traynor, P.: (Sp)iPhone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. In: Proceedings of the CCS'11 (2011)
19. Nandakumar, R., Iyer, V., Tan, D., Gollakota, S.: FingerIO: Using active sonar for fine-grained finger tracking. In: Proceedings of the CHI'16 (2016)

20. Nandakumar, R., Takakuwa, A., Kohno, T., Gollakota, S.: CovertBand: Activity information leakage using music. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, **1**(3), 87 (2017)
21. Narain, S., Sanatinia, A., Noubir, G.: Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning. In: Proceedings of the WiSec'14 (2014)
22. Pu, Q., Gupta, S., Gollakota, S., Patel, S.: Whole-home gesture recognition using wireless signals. In: Proceedings of the MobiCom'13 (2013)
23. Simon, L., Anderson, R.: PIN skimmer: inferring PINs through the camera and microphone. In: Proceedings of the SPSM'13 (2013)
24. Spreitzer, R., Moonsamy, V., Korak, T., Mangard, S.: Systematic classification of side-channel attacks: a case study for mobile devices. IEEE Commun. Surv. Tutor. **20**(1), 465–488 (2018)
25. Toreini, E., Randell, B., Hao, F.: An acoustic side channel attack on enigma. Computing Science Technical Report, Newcastle University (2015)
26. Turner, M.R.: Texture discrimination by Gabor functions. Biol. Cybern. **55**(2), 71–82 (1986). https://doi.org/10.1007/BF00341922
27. Uellenbeck, S., Dürmuth, M., Wolf, C., Holz, T.: Quantifying the security of graphical passwords: the case of android unlock patterns. In: Proceedings of the CCS'13 (2013)
28. Wang, W., Liu, A.X., Sun, K.: Device-free gesture tracking using acoustic signals. In: Proceedings of the MobiCom'16 (2016)
29. Wilson, J., Patwari, N.: See-through walls: motion tracking using variance-based radio tomography networks. IEEE Trans. Mob. Comput. **10**(5), 612–621 (2011)
30. Ye, G., Tang, Z., Fang, D., Chen, X., In: Kim, K., Taylor, B., Wang, Z. Cracking android pattern lock in five attempts. In: Proceedings of the NDSS'17 (2017)
31. Yun, S., Chen, Y.C., Zheng, H., Qiu, L., Mao, W.: Strata: fine-grained acoustic-based device-free tracking. In: Proceedings of the MobiSys'17 (2017)
32. Zhou, W., Zhou, Y., Jiang, X., Ning, P.: Detecting repackaged smartphone applications in third-party android marketplaces. In: Proceedings of the CODASPY'12 (2012a)
33. Zhou, Y., Wang, Z., Zhou, W., Jiang, X.: Hey, You, Get off of my market: detecting malicious apps in official and alternative android markets. In: Proceedings of the NDSS'12 (2012b)
34. Zhuang, L., Zhou, F., Tygar, J.D.: Keyboard acoustic emanations revisited. ACM Trans. Inf. Syst. Secur. (TISSEC) **13**(1), 3 (2009)